



Rsyslog

Table of contents

SEL and SOL Message Reception Format

Rsyslog Servers Configurations

IPMI Commands

Usage Examples

Setting Rsyslog Status of Two Configurations

Setting Rsyslog Status with IPv6 Address

Setting Rsyslog Status with Invalid Argument

Getting Rsyslog Status Information

Getting Non-existing Rsyslog Server Information

It is possible to dynamically configure rsyslog servers to receive system event log (SEL) messages and/or the BlueField SoC UART console printout (SOL) messages.

SEL and SOL Message Reception Format

SEL messages are received on the rsyslog server in the following format:

```
<Timestamp> <host> <EntryID-hex> | <Date> | <Time> | <Sensor-  
Type> | <Event-Type> | <Event-Direction> | <Description>
```

For example:

```
"2024-06-18T11:05:45.926095+03:00 ldev-platform-12-244.exam 75 |  
06/18/24 | 08:05:45 UTC | Voltage #0x08 | Lower Non-critical  
going low | Asserted"
```

SOL messages are received on the rsyslog server exactly as they appear in the BlueField console, including a timestamp and the hostname:

```
<Timestamp> <host> <message>
```

For example:

```
"2024-06-18T15:16:28.240538+03:00 ldev-platform-12-244  
systemd[1]: Starting RDMA Node Description Daemon"
```

Note

`$EscapeControlCharactersOnReceive` and `$Escape8BitCharactersOnReceive` should be turned off on the rsyslog server side.

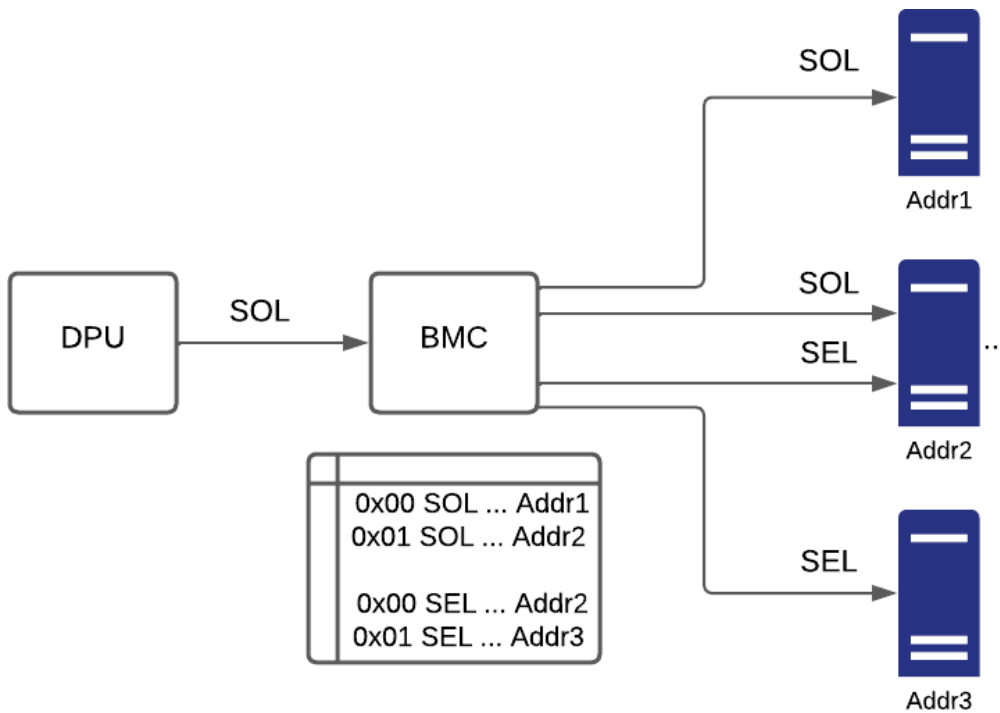
Rsyslog Servers Configurations

The rsyslog configurations define data streams. Each of them includes:

- Configuration identifier – An index (ranging from 0x00 to 0x09) AND a log type (SEL 0x01 or SOL 0x03)
- Status – Enable/disable
- Transport protocol – TCP/UDP
- Network protocol – IPv4/IPv6
- Server address
- Port

Note that configurations with the same index but different log types are considered to be different configurations. For example, 0x01-SOL and 0x01-SEL are distinct configurations.

The following diagram illustrates an example of three rsyslog servers receiving four data streams:



This setup requires four configurations:

- Configuration 0x00-SOL – Server1 receives SOL
- Configuration 0x01-SOL – Server2 receives SOL
- Configuration 0x00-SEL – Server2 receives SEL
- Configuration 0x01-SEL – Server3 receives SEL

Note

The BMC rsyslog configuration files located under `/etc/rsyslog.d` are automatically generated and are read-only. These files can only be modified using the IPMI commands listed later on this page.

IPMI Commands

The following table lists the IPMI commands for setting and getting rsyslog servers configurations:

net func	cmd	data	Description
0x32	0xD3	<Index> <LogType>	<p>Get rsyslog status – Displays information of the configured rsyslog server</p> <p>The request contains the index and the log type of the rsyslog server configuration, and it is 2 bytes long.</p> <p>The response contains the following information:</p> <pre><Index> <LogType> <Status> <TransportProtocol> <NetworkProtocol> <ServerAddress> <Port></pre> <ul style="list-style-type: none"> • Byte 1 – Completion code: <ul style="list-style-type: none"> ◦ 0x00 – Success (does not appear in IPMI textual response) ◦ 0x01 – Failure (the rest does not appear in IPMI response) • Byte 2 – Index <ul style="list-style-type: none"> ◦ Index of server (0x00-0x09) • Byte 3 – LogType <ul style="list-style-type: none"> ◦ 0x01 – SEL ◦ 0x03 – SOL • Byte 4 – Status <ul style="list-style-type: none"> ◦ 0x00 – Disabled ◦ 0x01 – Enabled • Byte 5 – Transport protocol <ul style="list-style-type: none"> ◦ 0x00 – UDP ◦ 0x01 – TCP • Byte 6 – Network protocol <ul style="list-style-type: none"> ◦ 0x00 – IPv4 ◦ 0x01 – IPv6 • Byte 7-n – Rsyslog server address <ul style="list-style-type: none"> ◦ Rsyslog addr (4/16 Bytes) • Byte n+1-n+2 – Port <ul style="list-style-type: none"> ◦ Rsyslog port. LSB first.

net func	cmd	data	Description
			<p>The response is 12 bytes long for IPv4 and 24 bytes long for IPv6.</p>
<p>0x32</p>	<p>0x0D4</p>	<p><Index> <LogType> <Status> <TransportProtocol> <NetworkProtocol> <ServerAddress> <Port></p>	<p>Set rsyslog status –</p> <ul style="list-style-type: none"> • Configures a new rsyslog server configuration if the configuration <Index> <LogType> does not exist. • Modifies an existing rsyslog server configuration if the configuration <Index> <LogType> does exist. <p>The command contains the following information:</p> <ul style="list-style-type: none"> • Byte 1 – Index <ul style="list-style-type: none"> ◦ Index of server (0x00-0x09) • Byte 2 – LogType <ul style="list-style-type: none"> ◦ 0x01 – SEL ◦ 0x03 – SOL • Byte 3 – Status <ul style="list-style-type: none"> ◦ 0x00 – Disabled ◦ 0x01 – Enabled • Byte 4 - Transport protocol <ul style="list-style-type: none"> ◦ 0x00 – UDP ◦ 0x01 – TCP • Byte 5 - Network protocol <ul style="list-style-type: none"> ◦ 0x00 – IPv4 ◦ 0x01 – IPv6 • Byte 6-n – Rsyslog server address <ul style="list-style-type: none"> ◦ Rsyslog addr (4/16 Bytes) • Byte n+1-n+2 – Port <ul style="list-style-type: none"> ◦ Rsyslog port. LSB first. <p>The command data is 11 bytes long for of IPv4 and 23 bytes log for IPv6. The response contains the completion code and is 1 byte long. The success completion code does not appear in IPMI textual response.</p>

Usage Examples

Setting Rsyslog Status of Two Configurations

The following commands create or modify two different rsyslog configurations with Index 0x00 and LogTypes SEL/SOL :

netfunc: 0x32, cmd: 0xD4, Indx: 0x00, LogType: 0x01(SEL) / 0x03(SOL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x01 0x01 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x03 0x01 0x01 0x00
0x0a 0xed 0x33 0xf4 0xfa 0x13
```

Now the same rsyslog server receives both SEL and SOL messages.

The following command disables the rsyslog configurations with Index 0x00 and LogTypes SOL:

netfunc: 0x32, cmd: 0xD4, Indx: 0x00, LogType: 0x03 (SOL), status: 0x00 (Disabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x00 0x03 0x00 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
```

Now the rsyslog server receives only SEL messages as the SOL configuration is disabled:

netfunc: 0x32, cmd: 0xD3, Indx: 0x00, LogType: 0x01(SEL) / 0x03(SOL)

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x01
00 01 01 01 00 0a ed 33 f4 fa 13
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x03
```



```
00 03 00 01 00 0a ed 33 f4 fa 13
```

Setting Rsyslog Status with IPv6 Address

The following command creates or modified an rsyslog configuration with an IPv6 address:

```
netfunc: 0x32, cmd: 0xD4, Indx: 0x07, LogType: 0x01 (SEL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x01 (IPv6) Address: 0xFD 0xFD 0xFD 0xFD 0x00 0x10 0x02 0x37 0x02 0x50 0x56 0xFF 0xFE 0x30 0x33 0xF4 (FDFD:FDFD:10:237:250:56FF:FE30:33F4) Port: 0xFA 0x13 (5114)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x07 0x01 0x01 0x01 0x01
0xfd 0xfd 0xfd 0xfd 0x00 0x10 0x02 0x37 0x02 0x50 0x56 0xff 0xfe
0x30 0x33 0xf4 0xfa 0x13
```

Setting Rsyslog Status with Invalid Argument

The following command attempts to create an rsyslog server configuration with an invalid index 0x0A (Valid indexes are 0x00-0x09):

```
netfunc: 0x32, cmd: 0xD4, Indx: 0x0A, LogType: 0x01 (SEL), status: 0x01 (Enabled), TP: 0x01 (TCP), NP: 0x00 (IPv4) Address: 0x0A 0xED 0x33 0xF4 (10.237.51.244) Port: 0xFA 0x13 (5114)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD4 0x0A 0x01 0x01 0x01 0x00
0x0A 0xED 0x33 0xF4 0xFA 0x13
Unable to send RAW command (channel=0x0 netfn=0x32 lun=0x0
cmd=0xd4 rsp=0xcc): Invalid data field in request
```

Getting Rsyslog Status Information

The following command displays the information of the rsyslog configuration with index 0 and LogType SEL :

```
netfunc: 0x32, cmd: 0xD3, Indx: 0x00, LogType: 0x01(SEL)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x00 0x01
00 01 01 01 00 0a ed 33 f4 fa 13
```

Getting Non-existing Rsyslog Server Information

The following command attempts to receive an information of a non-existing rsyslog configuration with index 0x06 and LogType SEL :

```
netfunc: 0x32, cmd: 0xD3, Indx: 0x06, LogType: 0x01(SEL)
```

```
root@dpu-bmc:~# ipmitool raw 0x32 0xD3 0x06 0x01
Unable to send RAW command (channel=0x0 netfn=0x32 lun=0x0
cmd=0xd3 rsp=0xcc): Invalid data field in request
```

Notice
This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality. NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete. NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document. NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk. NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the

application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Copyright 2024, NVIDIA. PDF Generated on 01/14/2025